Documentation for gnuPipe.h and gnuPipe.c

Steven Andrews, © 2003
See the document "LibDoc" for general information about this and other libraries.

```
FILE *gnuWindow();
void gnuKill(FILE *gnu);
void gnuCmd(FILE *gnu,char *s);
void gnuSetScales(FILE *gnu,float xa,float xb,float ya,float yb);
void gnuPrompt(FILE *gnu);
void gnuData(FILE *gnu,float *x,float *y,int m,int col);
void gnuData2(FILE *gnu,float *x,float *y,int *ct,int col);
```

Requires: `<stdio.h>`
Example program: `test.c`

Written 10/29/01 on Linux; no version on Macintosh. Some testing.

These routines allow minimal plotting of data by piping to the free gnuplot program. Apparently a version of gnuplot exists for the Macintosh, which I'll try to download sometime.

`gnuWindow` opens a pipe, returning a pointer to the file. This pointer needs to be sent back for all the other library routines.
`gnuKill` closes the pipe.
`gnuCmd` sends a command string to gnuplot without any prior filtering or processing. Since it's not possible to get messages back from gnuplot, there is no way for a calling program to know if an error occured.
`gnuSetScales` sets scales of the plot, where the *x* range is from `xa` to `xb` and the *y* range is from `ya` to `yb`. If `xa=xb` then autoscaling is used for `x`, and similarly for *y*.
`gnuPrompt` displays a prompt to the standard output to allow the program user to enter commands directly to gnuplot. It terminates when the user enters a blank line.
`gnuData` plots sets of *x*,*y* data to gnuplot, where there is one column of `x` values, with `m` items and `col` columns of `y` data, with `m` values each.
`gnuData2` is similar except that there are `col` columns of `x` data to plot against the respective columns of `y` data. Each column has `ct[]` elements.